Int. J. on Recent Trends in Engineering and Technology, Vol. 10, No. 1, Jan 2014

Experimental Analysis of an Energy-Efficient WSN

Ravi Kishore Kodali
National Institute of Technology, Warangal /Department of ECE, Warangal, India
Email: ravikkodali@gmail.com

***Abstract*— A wireless sensor network (WSN) is a collection of many tiny wireless sensor nodes which keep collecting various physical phenomena related data from their surrounding areas, where they are deployed and the same is sent towards the gateway. Different and many pieces of information, such as the occurrence of an event in the field and associated statistical values are computed from the collected data by these nodes. This data needs to be provided to authenticated end users, who are also remotely located, without any delays. This work presents an experimental WSN set up making use of accelerometer sensors and the experimental results of the same are presented. This work also presents the underlying *XMesh* routing protocol and an analysis of the its usage in the experimental set up is also provided.**

***Index Terms*— WSN, accelerometer, routing**

I. Introduction

Wireless sensor network (WSN) technology can be used to track and monitor various events in remote and hazardous areas [1],[2],[3]. A WSN consists of many nodes with limited resources and the nodes are used to carry out measurements of various physical phenomena related parameters of their surroundings and transmit the same towards the base station (BS). The nodes are deployed randomly in outdoor applications [4],[5],[6]. As the tiny nodes are deployed in remote areas, the resource constraints of the nodes such as limited computational capabilities and limited energy available with the nodes, make the design of WSN protocols a difficult task. In order to meet the requirements of various WSN applications, many routing and MAC layer protocols are proposed in the literature [7],[8].

In WSNs, the data is collected by each of the sensor nodes and the same is forwarded towards the gate way node. The gateway node connects the WSN to wireless /wired computer network. The gateway node, in turn, is connected the BS. A server which is connected to the BS using the Internet and the server is used to monitor and to control the data flow in the network. The same server stores the data from the network and the same is made available to the authenticated remote users through the Internet. The node tier comprises of all the WSN nodes deployed in the field. The IEEE 802.15.4 MAC protocol is used for communication among the nodes and the gateway node. A portable *Raspberry Pi* based computer board [9], powered by 5V/700mA power supply is connected to the gateway node. This board is Linux operating system platform and hosts a lightweight web server. This board also stores the data sent by the gateway node. The most recent data is made available to remote users through the web interface. The *Raspberry Pi* board is connected to 802.11n Wi-Fi module using USB port provides connectivity to the wireless local area network (WLAN). Further, it is connected to the Internet through wired local area network (LAN) and a proxy server. An authenticated remote user can access the data through the web interface of the *Raspberry-Pi* board. The data is also sent to the server equipped with a data logger along with a data parsing and a naming server. The clients receive the parsed data. The clients use their graphical user interface (GUI) to present the data and its analysis can be

DOI: 01.IJRTET.10.1.32

carried out. This work considers an event monitoring WSN application. The WSN application comprises of a static network with of static sensor nodes acting as relay nodes and mobile sensor nodes. *XMesh* [10], a dynamic multi-hop routing protocol is used. Various static nodes are connected to the gateway node. The mobile node sends its data to one of the nearby static nodes and the same is forwarded to the gateway node using the network formed by the static nodes. The static nodes receive the data from a mobile node and relay the same towards the BS. Further, this data is made available to the remote users through the BS. Analysis of this data is carried out using the *Moteview* GUI. The *Raspberry Pi* board facilitates in forwarding the data from the WSN to a remote user. An inexpensive *Raspberry Pi* board facilitates the data in the WSN to be made available on the Internet without requiring any dedicated computer system near the BS. This makes the concept of *Internet of things* (IoT) [11] easily realizable in an inexpensive manner. The remaining portion of the paper is organized as follows: The *XMesh* routing protocol is presented in section II. The hardware used in the experimental set up of WSN application is given in section III. The experimental setup is given in section IV, the experimental results are analysed in section V and the final section concludes the paper.

II. XMESH PROTOCOL

The network comprises of many WSN nodes. The nodes organize themselves and form a network making use of XMesh routing protocol. The XMesh protocol is an ad hoc, multi-hop routing protocol for WSNs [12],[13]. A WSN consists of many nodes and a BS. XMesh is self-healable and self-sustainable protocol, which supports dynamic addition and deletion of nodes at any instance [14]. XMesh uses mesh topology [15] using which nodes forward the data and establish an optimal multi-hop end-to-end path. Various routing paths are determined based on the link quality [16], which is computed by a receiving node using equation (1).

$$Link\ Quality = \frac{\text{No. of Recveived Messages}}{\text{No. of Expected Messages}} \tag{1}$$

The number of messages expected is extracted by tracking the sequence number of the message from a neighbouring mote. The link quality parameter is updat periodically A route from a parent node to a sink node is selected eliminating bad links. The multi-hop routing [17] supported by the *XMesh* protocol guarantees end-to-end connectivity and reduces the energy cost of a message transmission. The *RouteSelect* interface of the protocol selects an efficient route and the *RouteControl* interface monitors the route quality and alters the route state update interval. The Xmesh protocol can be programmed to operate in any of the three power modes: high power (HP), low power (LP) and extended low power (ELP). In both the HP and LP modes, all of the multi-hop mesh networking capabilities are enabled, while in ELP mode the nodes can not route the data and is connected to the gateway using single hop alone. The communication latency in LP mode is more as the devices on board are to conserve the limited battery energy. In the LP mode the status indication LEDs are turned off. The operating mode can be selected at the time of programming the node prior to deployment. For example, to force extended low power routing on the *iris* node, make command is: *make iris route, elp*

The application programs are developed using *nesC*. The BS uses the interfaces supported by the XMesh. The application code of the nodes differs from node to node as the nodes such as XMDA100, XMTS300, XMTS500 differ as these have different sensor types. Various time critical operations are written in the atomic section of the nesC application code, which can not be interrupted to avoid instability caused due to overwriting of the global variables. Further, when the timer is fired, sensor type data are measured. Various asynchronous events, such as, ADC data ready are implemented and the respective *getData()* function is called as and when an event happens. While transmitting a message along with the data, additional parameters like the node id, packet id and parent id are also transmitted. The *RouteControl* interface provides parent id. The *Send* interface selects the route and transmit the message. The *XCommand* interface handles the broadcast messages. For different commands like *SETRATE, SLEEP,WAKEUP,* appropriate actions are initiated. The application code for the BS forwards the packet for any XMesh application and inject XCommand packets into the network. The following command is used to build the BS application code on *iris* platform to operate 2.480 GHz frequency: *make iris base freq,2.480*

III. WSN HARDWARE

This work uses the IRIS sensor node with MTS400 sensor board. The MTS400 sensor board is installed on the mobile sensor node. This sensor board supports two-axis accelerometer which can be used for motion

detection. Also, the sensor board supports other sensor types humidity, temperature and light sensor to capture the surrounding environment.

The IRIS node comprises of Atmel's AT1281 micro-controller, AT86RF230 transceiver based on IEEE 802.15.4 and it is Zigbee compliant radio [18]. The node's RF transceiver can be tuned to different frequency channels in the ISM band 2.405 GHz to 2.48 GHz, with each separated by 5 MHz supporting 16 channels. The radio distance can be altered by varying the transmission power of the node from 3 dBm to -17.2 dBm. The IRIS node has 4 KB RAM and 128 KB flash memory to store the data and configuration parameters and other protocols. Tabel I provides the characteristics of the sensor board in MTS400 and Table II provides the IRIS mote specifications. The IRIS node also contains external 512 KB flash memory to support *On The Air Programming* (OTAP) [19] using which the node can be programmed post deployment. The programming board, MIB 520, is connected to one of the USB ports in the computer to program IRIS node. This board has an on-board ATMega16 controller to program the node. TinyOS operating system is to be installed on the computer which is used to program the nodes and connect the programming board to the computer. The TinyOS image along with the application code is downloaded to the node through the ATMega16 controller on the programming board. In order to support the programming feature using USB ports, FTDI USB Virtual COM port drivers need to be installed first.

TABLE I CHARACTERISTICS OF THE SENSOR BOARD- MTS400 [18]

Sensor type	Specification
Temperature	Accuracy: $\pm 2^o C$ Range: $-40^o C$ to $80^o C$
Light sensor	ON resistance (white light) $10K\Omega$ OFF Resistance $520K\Omega$
Pressure	Accuracy: ± 3.5 Range: 300 to 110 mbar
Accelerometer	Range: $\pm 2G$ $(G = 9.81 m/s^2)$ Sensitivity: 167 mV/G

TABLE II IRIS MOTE SPECIFICATIONS [21]

Parameter	Value
RAM	8 KB
ROM	128 KB
Clock Frequency	7.37 MHz
Supply voltage	2.7 to 3 V
TX Power	-17 to 3 dBm
TX current consumption	16 mA
RX current consumption	15 mA

The object under observation is tied with a node with MTS400 sensor board. The accelerometer sensor present on the board is used to track motion detection. Initially, the two axis accelerometer on the sensor board needs to be calibrated in order to detect the object movement with reduced error. The accelerometer is calibrated with respect to acceleration due to gravity ($g = 9.82\ m/s^2$) [21]. Towards this calibration, the X-axis of the accelerometer is directed towards the earth surface in a perpendicular manner and the reading of the accelerometer is noted down as $+1g$. The reading of accelerometer, when directed away from the surface of the earth is taken as $-1g$.

The same procedure is followed for the Y-axis calibration as well. Further, the readings obtained by both the axes of the accelerometer (acc_x, acc_y) are converted into the magnitude and angular values for the purpose of analysis. The magnitude (M) and angular (Θ) are computed using equation (2).

$$M = \sqrt{(acc_x)^2 + (acc_y)^2}$$
$$\theta = \tan^{-1}\left(\frac{acc_y}{acc_x}\right) \quad (2)$$

IV. EXPERIMENTAL SET UP

The *IRIS* sensor motes configured with XMesh protocol are deployed in the area under observation as given in Fig. 1. The deployed nodes are not in line of sight with the gateway. Multi-hop paths are established during the initial stage. After the network topology gets stabilised, the same is ready to monitor various events within the range of the network nodes. The object to be tracked is attached with a sensor node, which is capable of measuring two-axis acceleration, temperature, humidity and light conditions of the surroundings. If the node is in the radio range of the gateway device, it sends the data directly to the gateway. Otherwise, the mobile node attempts to establish a multi-hop path with the gateway through one of the neighbouring static nodes of the network. The mobile node is made to move along within the area of the network. The mobile node broadcasts its sensed data periodically. This data is collected by one of the neighbouring nodes of the network and the same is forwarded towards the gateway. This data is analysed with the *Moteview* software. The data is also sent to the *Raspberry Pi* board to facilitate access to the data to various authenticated users via the Internet.

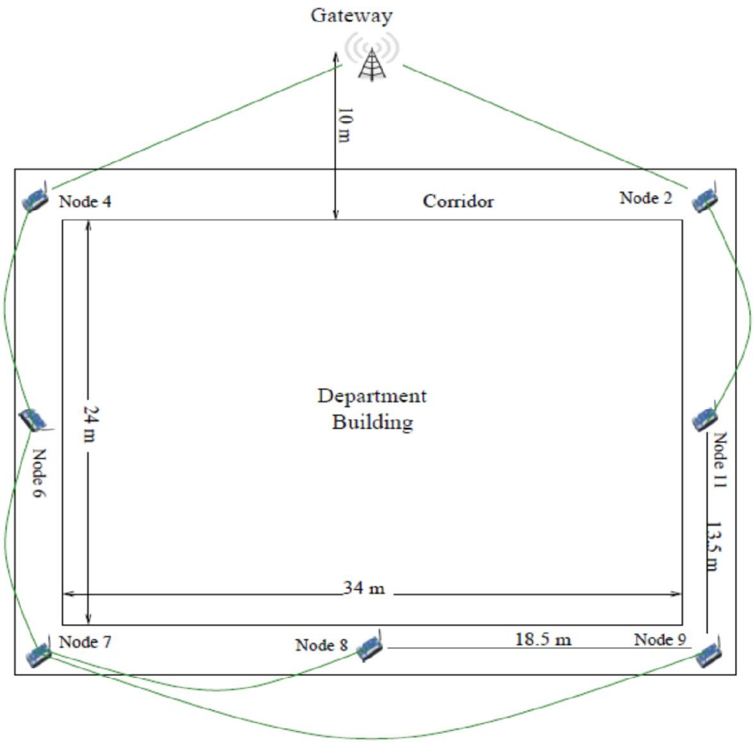


Figure 1: Wireless sensor network deployment [21]

V. EXPERIMENTAL RESULTS

Moteview is used to carry out the data analysis of the WSN. In this section, the experimental analysis using WSN hardware is presented. The XMesh protocol supports multi-hop routing to conserve the energy of nodes and to provide guaranteed connectivity between sensor motes and the gateway. To verify multi-hop routing capability of the XMesh protocol, initially all the nodes are brought nearer to gateway. Then, the node 4 is moved away from the gateway about > 30m in indoor conditions. Fig. 2 and Fig.3 show multi-hop connectivity. Now, node 3 is the parent node for the node 4. If two nodes are kept sufficiently long distance away from the gateway (node4 : 30m, node2 : 50m), a child node takes a total of three hops to get connected to the gateway. This experiment proves the reliability and the robustness of the XMesh protocol. But the time required to update the topology is on an average 1 minute. During this time any data collected cannot be forwarded to the gateway immediately.

The persistence of data collection from the mobile node depends on the robustness of routing protocol among the static sensor nodes. The response of XMesh routing protocol to the node failure is analysed by turning off

the critical node in the topology. Fig. 4 shows the network topology prior to turning off of the critical node. XMesh protocol is a dynamic routing protocol and hence parent nodes are altered frequently based on the link quality parameter. Based on the present topology, the gateway sends a command to the parent node having more number of child nodes to perform the data aggregation.

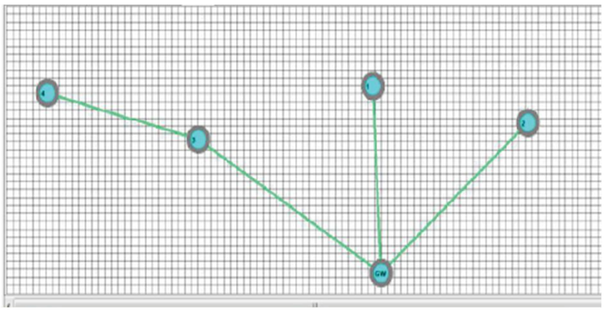


Figure 2: Multi-hop routing in WSN (two hops) [18]

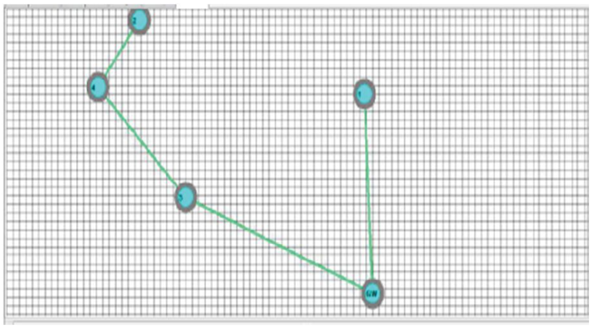


Figure 3: Mulit-hop routing in WSN (3 hops) [18]

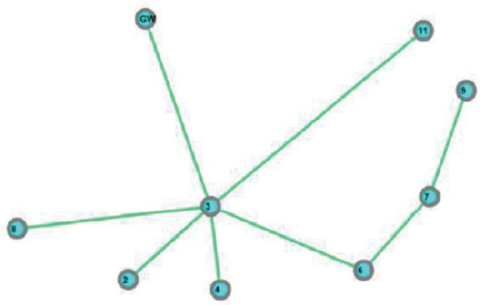


Figure 4: Faulty node detection in Moteview [18]

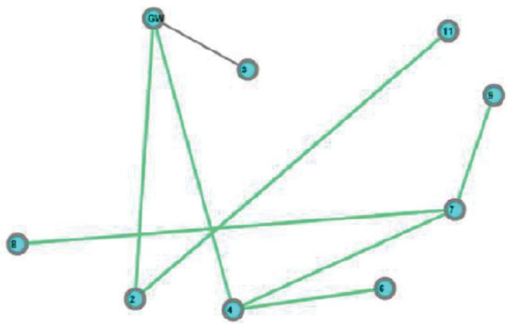


Figure 5: WSN multi-hop topology with node-3 disconnected [18]

Fig. 5 shows the topology when all the nodes except node 3 are active. *XCommand* interface provided by XMesh protocol is used to meet this purpose. After receiving the *XCommand* from the gateway, the parent node intercepts the data it is forwarding from its child nodes. The *intercept()* interface of the XMesh protocol is used instead of the *Receive()* interface. The change in parent node of the mobile node with respect to time is plotted in Fig. 6.

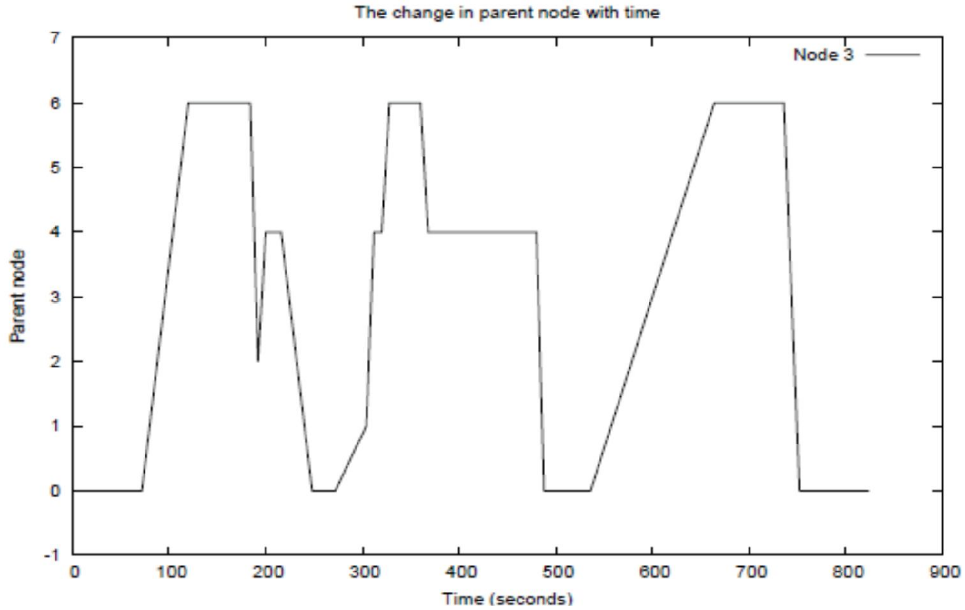


Figure 6: Update in parent node with respect to time

When the mobile node is not in the radio range it gets connected to the gateway through other mobile nodes. From the graph it can be seen that for most of the time the node is able to send the data to the gateway, even though it is not in its radio range. In this way, the area under surveillance is expanded without increasing the transmission power level. The feasibility of the network consisting of all the mobile sensor nodes with dynamic routing protocol is tested with the motion detection WSN application. The sensor board (MTS400) consist of two-axis accelerometer to detect the motion of the mobile node. The raw accelerometer data in x-axis direction and y-axis direction is plotted in figure 8. The sensor on the mobile node collects the data after every 8 seconds. The delay in the samples is due to the time required to update the topology.

A. Energy consumption analysis

The static nodes deployed in the network make use of XMesh protocol for routing the data towards the BS. XMesh can be operated in high power (hp) or low power (lp) mode as discussed in the section II. The energy consumption analysis for the *HP* and *LP* modes is given in Table III.

TABLE III ENERGY CONSUMPTION ANALYSIS [21]

Routing Mode	Depletion in voltage level of the node per hour
High power	42.9 mV
Low power	1.33 mV

The sampling rate and hence packet transmission rate is kept same for both the modes of routing (2seconds). In low power routing, the energy is conserved by reducing the channel listening period of the transceiver. The conservation of energy in the case of low power mode takes the cost of latency in topology formation and its update. When the nodes are moving in the area under surveillance, some of the nodes are not in the radio range of the gateway. In such a scenario, the data collected by the remote nodes are forwarded to the gateway through the nodes which are present in the radio range of the gateway at that point of time. The

parent node of each of the mobile node changes time to time. When at least one node is in the range of the gateway the data from all the mobile nodes can be gathered. The change in topology is analysed in Moteview software and the update in topology is presented in Fig. 7.

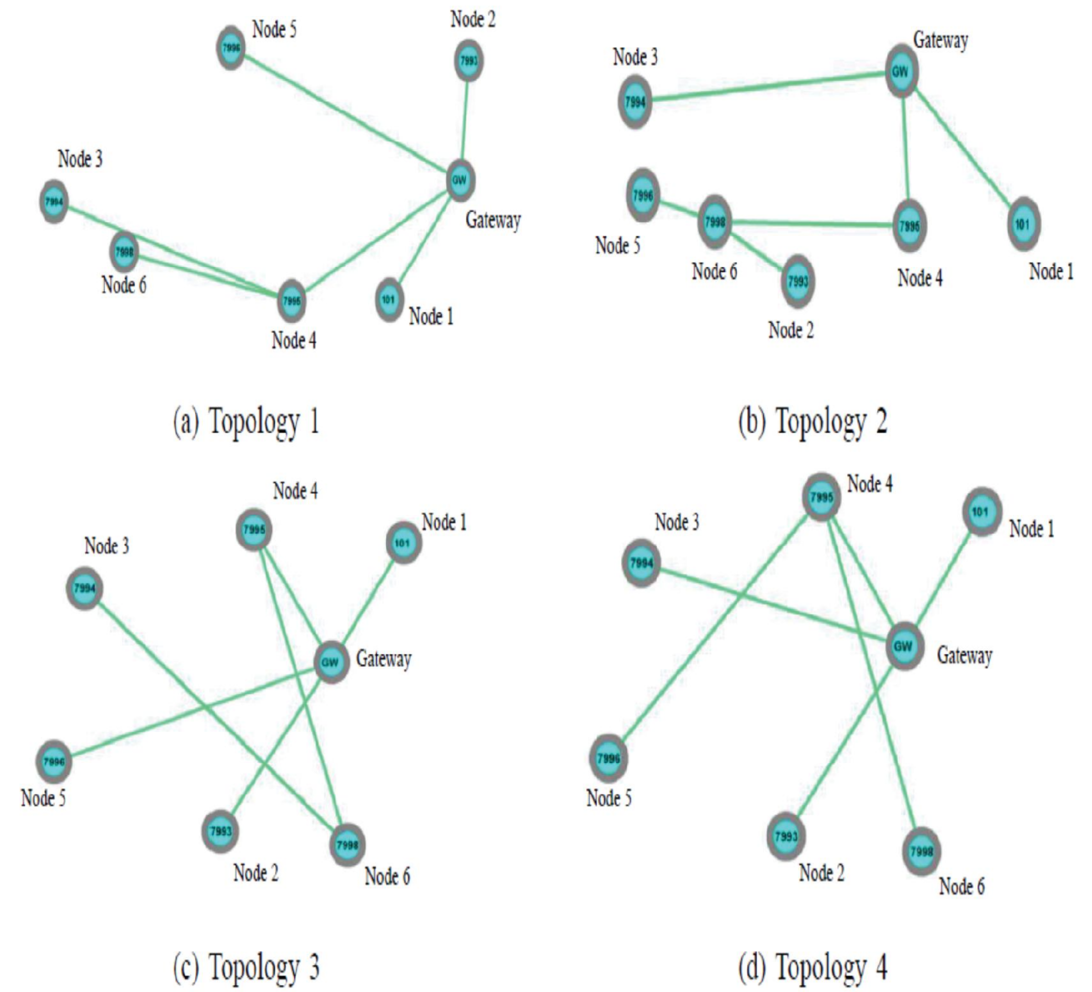
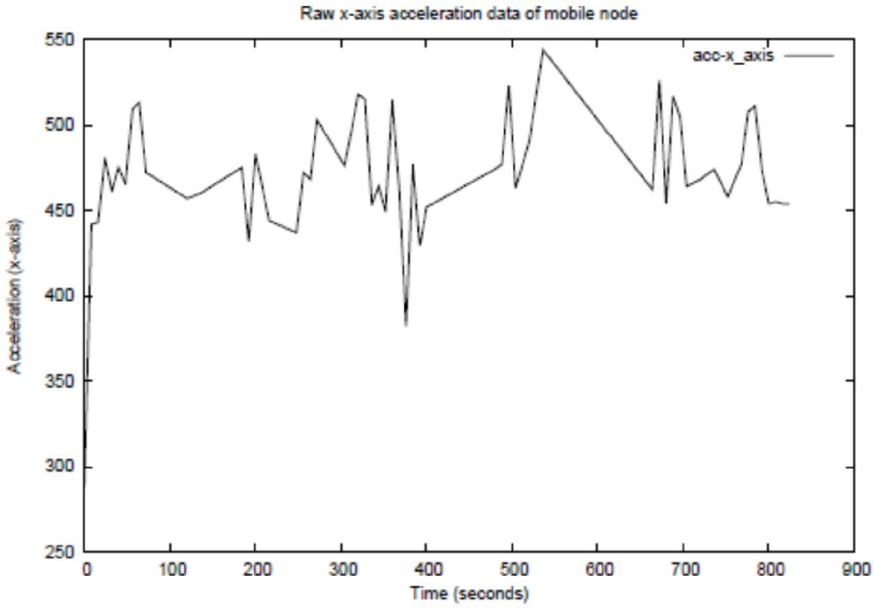
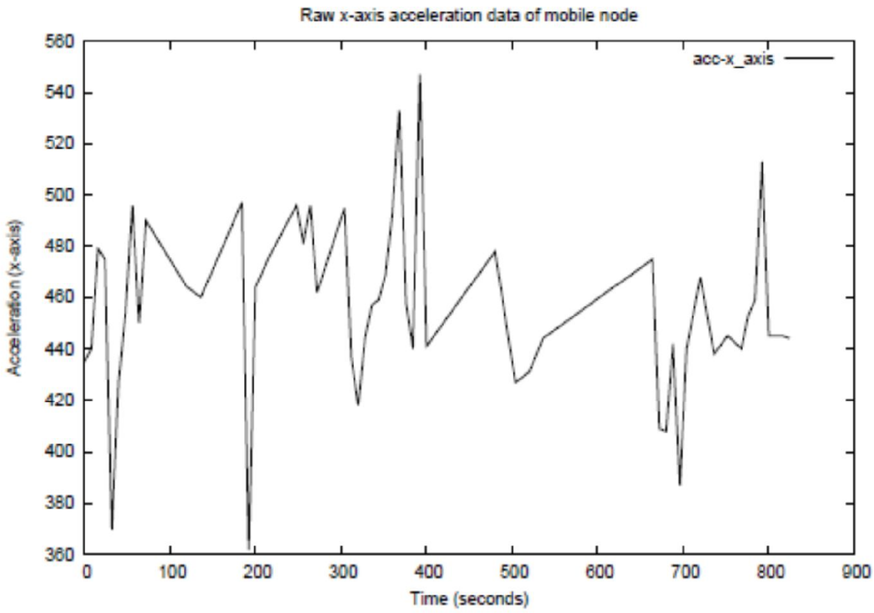


Figure 7: Update in topology

The movement of the mobile node is tracked by using its accelerometer readings. The raw accelerometer readings received by the gateway are shown in Fig. 8. When the mobile node is not in the range of the gateway, it passes the data by selecting the parent node from the nearby static nodes present in its communication range. Sometimes, the delay caused by the parent selection process causes certain amount of data loss, and the same is depicted in Fig. 8. From the magnitude and angle of the acceleration graphs the sudden movement and change in direction of the mobile object can be detected. The calibrated data is plotted in Fig. 9. The calibrated data is represented in terms of mg. The sensor can detect the magnitude values form +1g to −1g and the angle in the range of -90^0 to $+90^0$. The peaks in the magnitude graph show the sudden movement of the object and from the angle graph the direction of the motion can be detected. Using the magnitude and angle values calculated, the motion of the mobile object is traced. The calibrated accelerometer data, in terms of acceleration due to gravity is shown in Fig. 9c. The magnitude calculation assists in detecting sudden movements of the mobile node.



(a) Accelerometer raw data (x-axis)

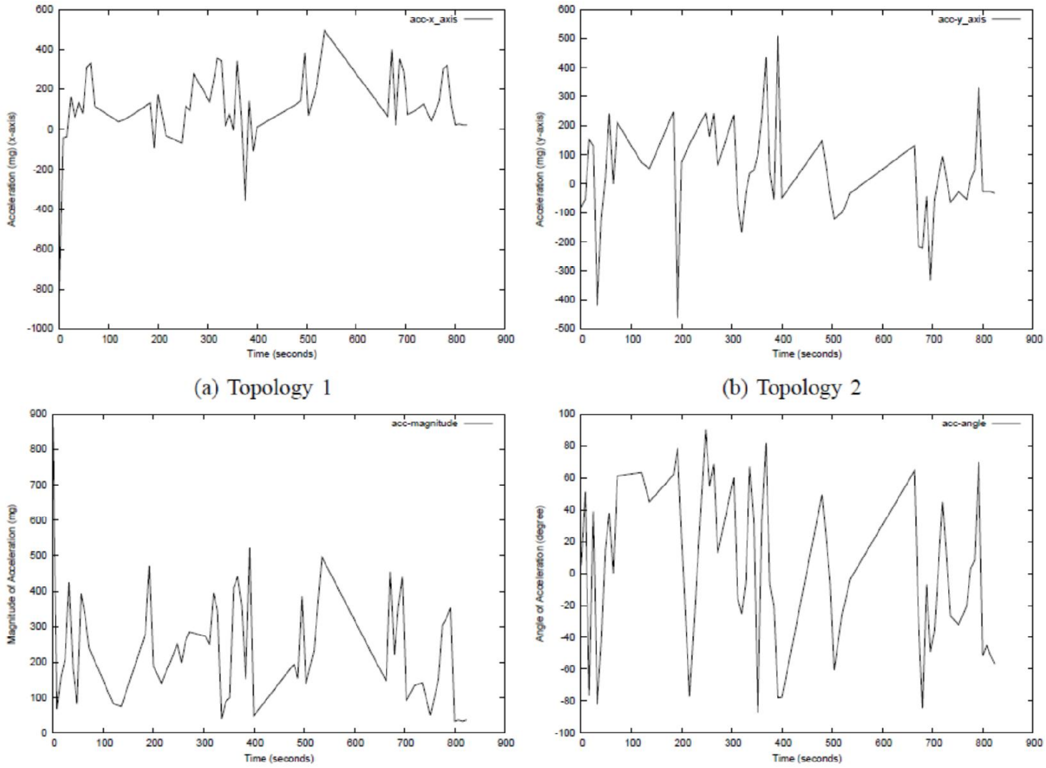


(b) Accelerometer raw data (y-axis)

Figure 8: Accelerometer raw data

VI. Conclusion

In this paper, an experimental WSN set up has been considered. The WSN setup, with the multi-hop routing protocol is discussed. An experimental analysis is carried out in this work, proves the feasibility of the proposed WSN architecture in real world applications. The depletion in the battery voltage level of a node corresponding to different operating modes of the Xmesh is given. The low power mode can be chosen when the object is moving slowly. When a node is made to operate its Xmesh in high power mode, the node depletes its energy sooner. To overcome this, data aggregation can be used. The data from the mobile node is



(a) Topology 1

(b) Topology 2

(c) Topology 3

(d) Topology 4

Figure 9: Calibrated acceleration values

made available to the gateway, even when the mobile node is not in the radio range of the gateway. However, it has been observed that the process selecting the parent by the mobile node causes latency and consequently results certain data loss.

REFERENCES

[1] K. Sohraby, D. Minoli, and T. Znati, Wireless sensor networks: technology, protocols, and applications. Wiley-interscience, 2007.

[2] A. Swami, Q. Zhao, Y.-W. Hong, and L. Tong, Wireless Sensor Networks: Signal Processing and Communications. Wiley, 2007.

[3] D. Angela, M. Ghenghea, and I. Bogdan, “Supporting environmental surveillance by using wireless sensor networks,” in Electrical and Electronics Engineering (ISEEE), 2010 3rd International Symposium on, 2010, pp. 216–219.

[4] A. Harun, M. F. Ramli, L. Kamarudin, D. Ndzi, A. Y. M. Shakaff, A. Zakaria, and M. Jaafar, “Comparative performance analysis of wireless rssi in wireless sensor networks motes in tropical mixed-crop precision farm,” in Intelligent Systems, Modelling and Simulation (ISMS), 2012 Third International Conference on, 2012, pp. 606–610.

[5] R. Yerra, A. Bharathi, P. Rajalakshmi, and U. Desai, “Wsn based power monitoring in smart grids,” in Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2011 Seventh International Conference on, 2011, pp. 401–406.

[6] T. Kuroiwa, M. Suzuki, Y. Yamashita, S. Saruwatari, T. Nagayama, and H. Morikawa, “A multi-channel bulk data collection for structural health monitoring using wireless sensor networks,” in Communications (APCC), 2012 18th Asia-Pacific Conference on, 2012, pp. 295–299.

[7] V. Kumar and S. Tiwari, “Performance of routing protocols for beaconenabled ieee 802.15.4 wsns with different duty cycle,” in Devices and Communications (ICDeCom), 2011 International Conference on, 2011, pp. 1–5.

[8] O. Adewumi, K. Djouani, and A. Kurien, “Rssi based indoor and outdoor distance estimation for localization in wsn,” in Industrial Technology (ICIT), 2013 IEEE International Conference on, 2013, pp. 1534–1539.

[9] E. Upton and G. Halfacree, Meet the Raspberry Pi. Wiley, 2012.

[10] X. U. Manual and D. Revision, “Crossbow technologies,” Milpitas, CA, Apr, 2007.

[11] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010.
[12] T. W. Davis, X. Liang, M. Navarro, D. Bhatnagar, and Y. Liang, “An experimental study of wsn power efficiency: Micaz networks with xmesh,” International Journal of Distributed Sensor Networks, vol. 2012, 2012.
[13] A. Brokalakis, G.-G. Mplemenos, K. Papadopoulos, and I. Papaefstathiou, “Resense: An innovative, reconfigurable, powerful and energy efficient wsn node,” in Communications (ICC), 2011 IEEE International Conference on, 2011, pp. 1–5.
[14] A. Teo, G. Singh, and J. McEachen, “Evaluation of the xmesh routing protocol in wireless sensor networks,” in Circuits and Systems, 2006. MWSCAS’06. 49th IEEE International Midwest Symposium on, vol. 2. IEEE, 2006, pp. 113–117.
[15] S. Waharte, R. Boutaba, Y. Iraqi, and B. Ishibashi, “Routing protocols in wireless mesh networks: challenges and design considerations,” Multimedia Tools and Applications, vol. 29, no. 3, pp. 285–303, 2006.
[16] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” Wireless Networks, vol. 11, no. 4, pp. 419–434, 2005.
[17] M. Conti and S. Giordano, “Multihop ad hoc networking: The reality,” Communications Magazine, IEEE, vol. 45, no. 4, pp. 88–95, 2007.
[18] R. K. Kodali, “Experimental analysis of an event tracking energy-efficient wsn,” in Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on, 2013, pp. 1293–1298.
[19] Z. Alliance, “Zigbee specification,” ZigBee Document 053474r13, pp. 344–346, 2006.
[20] N. Bin Shafi, K. Ali, and H. Hassanein, “No-reboot and zero-flash over the-air programming for wireless sensor networks,” in Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on, 2012, pp. 371–379.
[21] R. K. Kodali and N. Narasimha Sarma, “Experimental wsn setup using xmesh networking protocol,” in Advanced Electronic Systems (ICAES), 2013 International Conference on, 2013, pp. 267–271.